Springer International Journal of Social Robotics manuscript No.
(will be inserted by the editor)

A Component-based Approach to Visual Person Tracking from a Mobile Platform

Simone Frintrop[1] · **Achim Königs**[2] · **Frank Hoeller**[2] · **Dirk Schulz**[2]

Received: date / Accepted: date

Abstract In this article, we present a component-based visual tracker for mobile platforms with an application to person tracking. The core of the technique is a component-based descriptor that captures the structure and appearance of a target in a flexible way. This descriptor can be learned quickly from a single training image and is easily adaptable to different objects. It is especially well suited to represent humans since they usually do not have a uniform appearance but, due to clothing, consist of different parts with different appearance. We show how this component-based descriptor can be integrated into a visual tracker based on the well known Condensation algorithm. Several person tracking experiments carried out with a mobile robot in different laboratory environments show that the system is able to follow people autonomously and to distinguish individuals. We furthermore illustrate the advantage of our approach compared to other tracking methods.

Keywords Visual Tracking · Component-based Tracking · Person Tracking

1 Introduction

An important skill for mobile service robots is the ability to detect and keep track of individual humans in their surrounding. Especially robots that are designed to provide services to individual persons need to be able to distinguish their client from the surrounding environment. Usually, such systems shall be able to learn the appearance of a target person quickly, possibly from a single snapshot. Additionally, to run on a mobile platform the approaches have to be real-time capable and robust to illumination changes, motion blur and quick viewpoint changes.

S. Frintrop
Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität,
53117 Bonn, Germany.
E-mail: frintrop@iai.uni-bonn.de

A. Königs, F. Hoeller, D. Schulz
Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE),
53343 Wachtberg, Germany.
E-mail: {koenigs,hoeller,schulz}@fgan.de

While many approaches have been proposed to track humans, most of them are not designed to distinguish individuals. This is especially true for laser-based systems that usually track the legs of people, or for model-based vision approaches that consider the shape of objects. Well suited to distinguish people are feature-based vision approaches. For these methods, it is especially important to detect discriminative features that distinguish the target well from the background. However different parts of complex objects, such as people, provide different discriminability from the background. If a person wears, for example, a shirt in a color similar to the background, it has a low discriminability while the trousers on the other hand might have a high discriminability. A good feature descriptor shall consider this variable discriminability and focus on the most discriminative parts. Since the structure of parts differs from target to target, it is preferable to automatically detect the different parts instead of using a rigid template.

In this paper, we present a component-based approach to visual tracking that is able to automatically detect the most discriminative parts of a target person and to quickly learn its appearance from a single frame. Depending on the appearance of the person (clothing, hair color, skin color etc.), the system determines a flexible number of components, each representing a discriminative part with respect to a certain feature channel. The resulting components form a target template that is used in the following frames to detect the most likely target position. A similarity measure determines the similarity between the target template and image regions in the following frames. Instead of computing the similarity for each pixel, we employ the component-based approach within a Condensation-based person tracker [20]. For this purpose, the similarity measure is converted to a likelihood function that is used as the observation model within the particle filter.

This approach leads to a robust and flexible tracker that is quickly applicable to track arbitrary people in unknown environments. It is able to work in real-time on a mobile platform. We evaluated the approach in different settings: first, we compared the approach to other color-based tracking approaches and show that the performance of the component-based tracking outperforms the other approaches considerably. Second, we tested the ability of the system to distinguish a target person from other people that cross their way in front of the robot. Finally, we showed that the robot is able to follow a person autonomously in different settings of our laboratory environment under varying lighting conditions and backgrounds.

The remainder of the article is organized as follows. After discussing related work in Section 2, we introduce the component-based descriptor in Section 3. In Section 4, we explain the visual tracking system. Section 5 briefly explains how the approach is integrated into a prototypical person following application and presents experimental results. We finally conclude in Section 6.

2 Related work

In mobile robotics, researchers have developed person tracking techniques for different sensors. A frequently used approach is to use laser range finders, as these sensors are available on many robots for collision avoidance purposes. Because laser sensors usually only provide distance information to objects in the environment, most laser-based approaches only keep track of the motion of people and do not try to distinguish between individuals [25, 31]. However, several techniques have been developed that utilize the appearance of a person's legs in the data, to reduce the risk of track loss or

the confusion of tracks of different persons. For example, Arras et al. [2] use AdaBoost to train a detector for the legs of persons in laser range profiles and in more recent work [3] they suggest a two-leg constraint in combination with a specialized occlusion handling technique to increase the robustness against track loss. Taylor and Kleeman [34] use a switched dynamic model to even track the repetitive leg motion for this purpose.

Other authors improve the robustness of laser-based tracking by additionally taking camera information into account. Using this combination of sensors, the spatial tracking can still be performed on the laser data, while the camera immediately provides informative appearance information to distinguish between persons. For example, Bennewitz et al. [5] and Bellotto and Hu [4] use color histograms to discriminate between the persons being tracked. Schulz [30] uses a shape matching approach to distinguish between persons; a probabilistic exemplar approach is applied to track characteristic silhouettes of individuals over time. However, this requires a time-consuming learning process for the exemplar model of each new person.

In machine vision, people tracking is a well-studied problem. Two main approaches can be distinguished: *model-based* and *feature-based* methods. In model-based tracking, a model of the object is learned in advance, usually from a large set of training images which show the object from different viewpoints and in different poses [29]. Learning a model of a human is difficult because of the dimensionality of the human body and the variability in human motion. Current approaches include simplified human body models, e.g. stick, ellipsoidal, cylindric or skeleton models [8, 37, 24], or shape-from-silhouettes models [9]. While these approaches have reached good performance in laboratory settings with static cameras, they are usually not applicable in real-world environments on a mobile system. They usually do not operate in real-time and often rely on a static, uniform background. A model-based approach that works from moving cameras is shape matching. For example, Gavrila [17] suggests an exemplar-based technique that employs fast Chamfer matching to detect the shapes of pedestrians in images in real-time. The technique has been adopted for a particle filter tracking algorithm by Toyama and Blake [36]. However, it is not possible to adapt the rather large exemplar models on-line and, thus, the approach is not capable of distinguishing between persons during tracking. A modeling technique related to exemplars are implicit shape models [22] which, in comparison to pure exemplar approaches, improve the robustness against partial occlusions of objects. However, these models can also not be adapted online and are generally also not suitable to distinguish individual people. The final model-based technique, we want to mention, is tracking-by-detection, which has become increasingly popular over the last years. Typically, these approaches learn classifiers based on feature descriptors in order to detect and track humans in images [12, 1, 38]. Due to carefully chosen object specific feature sets, very reliable detections are achieved that can directly be used as observations within a tracking algorithm. The combination of part detectors even allows for partial occlusions. However, the classifiers generally require an off-line learning phase on a rather large training set. Our descriptor, in contrast, does not allow to detect people, but is used to acquire a robust observation model for individual objects from a single image for tracking. On-line supervised learning techiques can be applied to train classifiers for a similar purpose [18, 32], but need a larger image sequence to acquire the models.

Feature-based tracking approaches on the other hand do not learn a model but track an object based on simple features such as color cues or corners. One approach for feature-based tracking is the Mean Shift algorithm [10, 11] which characterizes objects by their color distribution. The algorithm tracks objects by carrying out a gradient

descent in the image that minimizes the dissimilarity between the local color statistics in the image and the object's color histogram. An extension of this method is the CamShift algorithm [7]. Other groups integrate color histograms into a particle tracker [27,28]. In previous work, we have used a cognitive observation model for visual tracking that was based on features inspired by human visual perception [14,16]. Several ideas from this work have been integrated into the current approach. Over the last years, techniques which use interest points, like colored Harris corners [23] or SIFT features [33] for object tracking have been introduced. Note that these approaches usually rely on textured objects and a certain image resolution and quality to work well. While these feature-based techniques are not especially designed for person tracking, they are commonly applied in this area.

Some people have also suggested to store different representations for different parts of the objects. For example, Pérez et al. determine different color histograms for different, rigidly linked parts of the target [27,28]. Beuter et al. train a top-down attention model to learn the face and the torso of a person separately [6]. We are however not aware of any work that determines the number and kinds of components of a target automatically to obtain a flexible descriptor as we will present in this work.

3 A Multi-Component Target Descriptor

In this section, we introduce the multi-component descriptor that represents a target. The computation consists of two steps. First, six intensity and color feature maps are computed (sec. 3.1), second, components are determined within the feature maps and combined to form the descriptor (sec. 3.2). Finally, we describe how the descriptor is matched to a region in a new frame to test if the target is present (sec. 3.3).

3.1 Feature map computations

In this section, we describe the computation of six intensity and color maps as a basis for the component-based descriptor. The computation of these feature maps is based on concepts from the human visual system in which color opponent cells determine the contrast of a center region and its surround [26]. The computation is the same as in the visual attention system VOCUS [13,15] and similar to Itti's attention system NVT [21].[1] An overview of the processing is displayed in Fig. 1.

First, the input image is converted to an image in the CIELAB color space (also $L^*a^*b^*$), smoothed with a Gaussian filter and subsampled twice to reduce the influence of image noise. We call the resulting image I_{Lab}. The CIELAB space has the dimension L for lightness and a and b for the color-opponent dimensions; it is perceptually uniform, which means that a change of a certain amount in a color value is perceived as a change of about the same amount in human visual perception. Furthermore, the space suits our purpose especially well since the four main colors red, green, blue and yellow are at the end of the axes a and b. This will show to be useful for our computations. Each of the 6 ends of the axes that confine the color space serves as one prototype color, resulting in two intensity prototypes for white and black and four color prototypes for red, green, blue, and yellow (cf. Fig. 1, left, top right corner).

[1] Differences to NVT include the use of a different color space and of integral images to speed up processing; more differences outlined in [13].

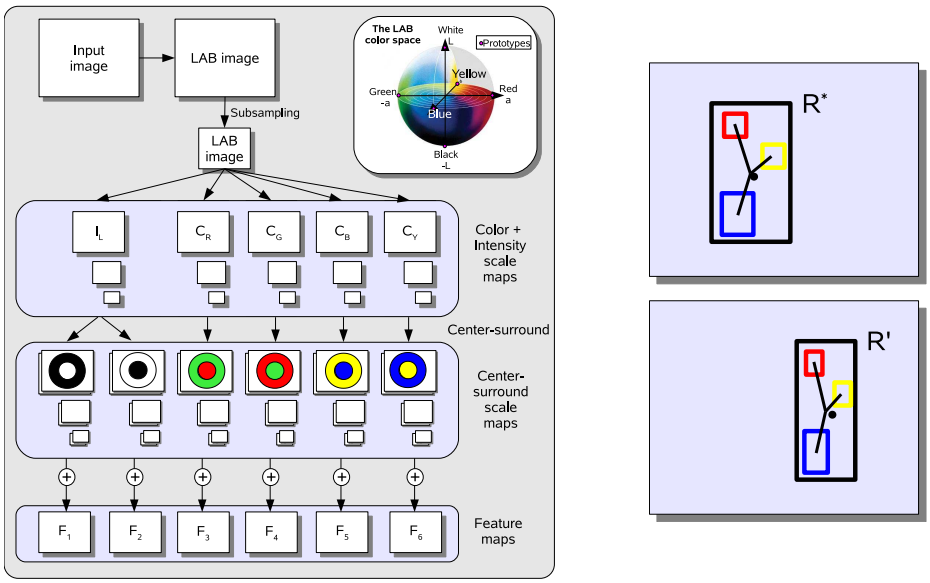

Fig. 1 Left: The feature computations: from an input image, 6 feature maps are computed, showing bright-dark, dark-bright, red-green, green-red, blue-yellow, and yellow-blue contrasts. Right, top: An illustration of the template $\mathbf{M}_{\mathbf{R}*}$ for the target region $\mathbf{R}^*$. The three colored rectangles denote the $m_{i,j}$; the different colors illustrate the feature maps they result from. Right, bottom: the template $\mathbf{M}_{\mathbf{R}'}$ adapted to region $\mathbf{R}'$.

Then, the computation of feature maps is started. We treat intensity and color computations separately since this results in a higher illumination invariance. The intensity computations can be performed directly from the L channel I_L. The color computations are performed on the color layer I_{ab} spanned by a and b. Now, we determine four color specific maps C_i that represent the four colors red, green, blue and yellow.

For each of the color maps C_i, there is one prototype color P_i (cf. Fig. 1, left, top right corner) and each pixel $C_i(x, y)$ in a color map stores the Euclidean distance to the corresponding prototype color P_i:

$$C_i(x, y) = V_{max} - ||I_{ab}(x, y) - P_i|| \qquad i \in \{1, ..., 4\}, \tag{1}$$

where $V_{max} = 255$ is the maximal pixel value and the prototypes P_i are the ends of the a and b axes with coordinates $(0, 127), (127, 0), (255, 127), (127, 255)$ in an 8-bit I_{ab}.

Next, image pyramids with 3 levels are determined from I_L and C_i. This enables flexibility to scale changes. On each of these scale maps in the pyramids we perform *center-surround mechanisms*. These are filters that detect image contrasts between a center c and a surround region s. Applied to our scale maps, the filters detect intensity and color contrasts. On the color maps, the filters react especially strong to red-green, green-red, blue-yellow, and yellow-blue contrasts. We use surround regions of two different sizes (radius 3 and 7 pixels, center 1 pixel), resulting in six center-surround maps $S_{i,j}, j \in \{1, ..., 6\}$ for each color/intensity (details in [13]). Note that center surround applied to the intensity scale maps detects only bright-dark contrast. To additionally determine dark-bright contrasts, we compute the opposite difference $s - c$. To speed up processing, all center-surround filters are computed with integral images [15].

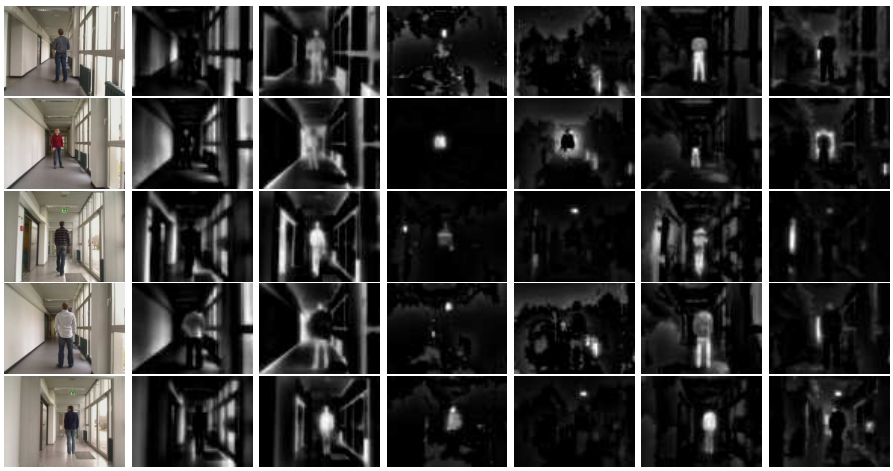

Fig. 2 The initial frames used for the experiments in sec. 5.1 and corresponding feature maps.

Finally, we sum up the 36 center-surround scale maps to obtain 6 feature maps F_i: $F_i = \sum_{j=1}^{6} S_{i,j}$. The feature maps for some example images are displayed in Fig. 2.

3.2 Determining a target descriptor

The target descriptor consists of components that have a strong contrast within a certain feature dimension. It is derived from the feature maps. A component is a peak in one of the feature maps within the target region $\mathbf{R^*} = (x^*, y^*, w^*, h^*)$. The peaks are detected by first detecting local intensity maxima and then segmenting the region around the maxima with region growing. For easier computations, the regions are approximated by rectangular bounding boxes that we call $m_{i,j}$, where i denotes the feature map and j the different maxima in a map. Hereby, the number of components per map is flexible and depends on the appearance of the object. Additionally, we add the whole target region as one of the $m_{i,j}$ to make the descriptor more robust.

The positions of the regions $m_{i,j}$ are stored relative to the center of $\mathbf{R^*}$ and represent a template $\mathbf{M_{R^*}} = \{m_{i,j} | i \in \{1,..,6\}, j \in \{1,..,l_i\}\}$ (cf. Fig. 1, right top, and Fig. 5). Now, we compute a descriptor vector from the $m_{i,j}$. For each $m_{i,j}$, we compute the ratio of the mean intensity value within $m_{i,j}$ and the mean value of the background:

$$\rho_{i,j} = \frac{mean(m_{i,j})}{mean(F_i \backslash m_{i,j})} \tag{2}$$

The mean is computed with integral images, to speed up processing and enable constant computation times for each region, independent of the size of the region. Thus, the target descriptor that we obtain is $\mathbf{d^*} = \{\rho_{i,j} | i \in \{1,..,6\}, j \in \{1,..,l_i\}\}$.

3.3 Matching the descriptor to an image region

In order to match the target descriptor $\mathbf{d^*}$ to an image region $\mathbf{R'}$ of arbitrary size and dimensions, we first determine the factors f_w and f_h that represent the difference

in size between the target region $\mathbf{R^*}$ and $\mathbf{R'}$: $f_w = R'_w/R^*_w, \quad f_h = R'_h/R^*_h$, where R'_w, R^*_w denote the width and R'_h, R^*_h the height of the regions. Now, an adapted template $\mathbf{M_{R'}}$ is computed by extending or compressing all $m_{i,j} \in \mathbf{M_{R^*}}$ with f_w and f_h: $m'_w = f_w * m^*_w, \quad m'_h = f_h * m^*_h, \quad \forall m' \in \mathbf{M_{R'}}, m^* \in \mathbf{M_{R^*}}$ (cf. Fig. 1, right bottom). $\mathbf{M_{R'}}$ is now used to compute a descriptor $\mathbf{d'}$ equivalently as in eq. 2.

Finally, the descriptors $\mathbf{d^*}$ and $\mathbf{d'}$ are matched by computing the similarity of the vectors. As similarity measure, we use the Tanimoto coefficient:

$$T(\mathbf{d}^*, \mathbf{d}') = \frac{\mathbf{d}^* \cdot \mathbf{d}'}{||\mathbf{d}^*||^2 + ||\mathbf{d}'||^2 - \mathbf{d}^* \cdot \mathbf{d}'}. \tag{3}$$

The Tanimoto coefficient produces values in the interval $[0, 1]$, the higher the value the higher the similarity. If the two vectors are identical, the coefficient is 1.

4 The Visual Tracking System

The tracking system we present uses the component-based descriptor from Sec. 3 for the observation model of a particle filter. It employs the standard Condensation algorithm [20] which maintains a set of weighted particles over time using a recursive procedure based on three steps: First, the system draws particles randomly from the particle set of the previous time step, where each particle is drawn with a probability proportional to the associated weight of the particle. Second, the particles are transformed (predicted) according to a motion model. Finally, all particles are assigned new weights according to an observation model and the object state is estimated.

Let us first introduce the notation. At each point in time $t \in \{1, .., T\}$, the particle filter recursively computes an estimate of the probability density of the person's location within the image using a set of J (here $J = 500$) particles $\mathbf{\Phi}_t = \{\phi^1_t, ...\phi^J_t\}$ with $\phi^j_t = (\mathbf{s}^j_t, \pi^j_t, \mathbf{d}^j_t), \quad j \in \{1, ..., J\}$. Here, $\mathbf{s}^j_t = (x, y, v_x, v_y, w, h)$ is the state vector that specifies the particle's region with center (x, y), width w and height h – in the following, the region is also denoted as $\mathbf{R}^j_t = (x, y, w, h)$. v_x and v_y specify the current velocity of the particle in x and y directions. Each particle additionally has a weight π^j_t determining the relevance of the particle with respect to the target, and the component-based descriptor $\mathbf{d}^j_t$ that describes the appearance of the particle region.

In the following, we first mention how the system is initialized (sec. 4.1), second describe the motion model (sec. 4.2), and finally, specify the observation model as core of the system (sec. 4.3).

4.1 Initialization

To start the tracking process, the initial target region $\mathbf{R}^*$ has to be specified in the first frame. This can be carried out manually or automatically with a separate detection module. Here, we initialize manually. Based on the initial target region $\mathbf{R}^*$, the component-based descriptor $\mathbf{d}^*$ is computed that describes the appearance of the person. The initial particle set $\mathbf{\Phi}_0 = \{(\mathbf{s}^j_0, \pi^j_0, \mathbf{d}^j_0) \,|\, j = 1, ..., J\}$ is generated by randomly distributing the initial target location around the region's center (x^*, y^*). The velocity components v_x and v_y are initially set to 0 and the region dimensions of each particle are initialized with the dimensions of $\mathbf{R}^*$. The particle weights π^j_0 are set to $1/J$.

4.2 Motion model

The object's motion is modeled by a simple first order autoregressive process in which the state $\mathbf{s}_t^j$ of a particle depends only on the state of the particle in the previous frame:

$$\mathbf{s}_t^j = \mathbf{M} \cdot \mathbf{s}_{t-1}^j + \mathbf{Q}. \tag{4}$$

Here, $\mathbf{M}$ is a state transition matrix of a constant velocity model and $\mathbf{Q}$ is a random variable that denotes some white Gaussian noise. This enables a flexible adaption of position and size of the particle region as well as of its velocity.[2] Thus the system is able to quickly react to velocity changes of the object.

4.3 Observation model

In visual tracking, the choice of the observation model is the most crucial step since it decides which particles will survive. It therefore has the strongest influence on the estimated position of the target. Here, we use the component-based descriptor to determine the feature description for the target and for each particle, enabling the comparison and weighting of particles.

First, we compute a descriptor $\mathbf{d_t^j}$ for each of the particles according to sec. 3.2. That means, the target template $\mathbf{M_{R^*}}$ is adapted to the size of the current particle and the descriptor $\mathbf{d_t^j}$ is computed for the resulting template $\mathbf{M_t^j}$. Then, the weight of a particle is computed based on the Tanimoto coefficient as

$$\pi_t^j = c \cdot e^{\lambda \cdot T(\mathbf{d}^*, \mathbf{d}_t^j)}. \tag{5}$$

This function prioritizes particles which are very similar to $\mathbf{d}^*$ by assigning an especially high weight. A value of $\lambda = 14$ has shown to be useful in our experiments. The parameter c is a normalization factor which is chosen so that $\sum_{j=1}^{J} \pi_t^j = 1$.

Finally, the current target state, including target position and size, can be estimated as a weighted average of the particles by

$$\mathbf{x}_t = \sum_{j=1}^{J} \pi_t^j \cdot \mathbf{s}_t^j. \tag{6}$$

5 Experiments and Results

The experiments were carried out using a RWI B21 robot equipped with a simple USB web camera mounted on a pantilt unit (see Fig. 3, left). The camera captures 15 frames/sec, with a resolution of 320×240. The software runs on a 2GHz dual core PC onboard the robot. For the experiments, the tracking application was implemented within the software framework RoSe developed at FKIE [35]. This framework consists of roughly 30 modules which exchange information over a UDP-based communication infrastructure. It is specifically designed to allow for the easy assembly of multi-robot applications, which extensively use wireless ad-hoc communication. However, here we only required two modules on a single robot:

[2] The size of the region is not adapted by M but only by Q.

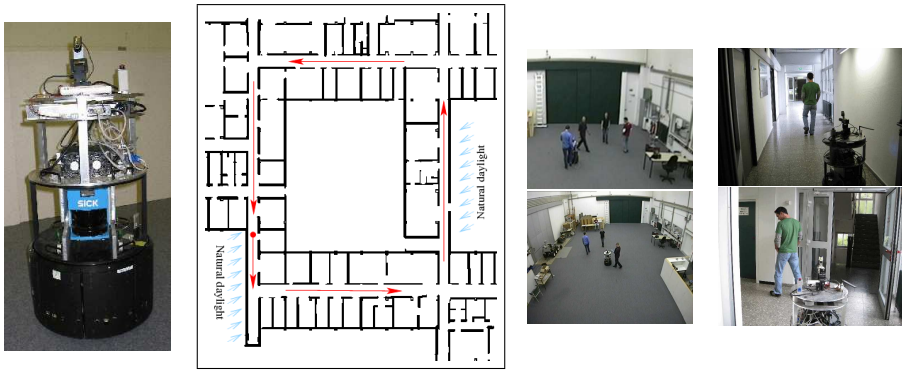

Fig. 3 Left: the RWI B21 robot *Blücher*. The images were taken using the small pantilt mounted webcam on top of the robot. Middle: An outline of the FKIE hallway environment. The red arrows indicate the corridors. Right: Experiments in our robot experimentation hall and in the corridors.

1) A visual tracking module, which captures the images and employs the tracking algorithm for tracking a single person within the image. Based on the pixel location of the person computed by the vision-based tracker, the module computes a heading direction relative to the robot, steers the pantilt unit in order to center the person within the image and commands the robot to follow the person. This is achieved by continuously instructing the reactive collision avoidance component of the robot to drive to a goal location a few meters ahead, in the direction of the moving person.

2) The collision avoidance component of the robot. It is specifically designed for the task of following moving persons based on motion tracking information. It does so by applying an expansive spaces tree algorithm, which carries out a search for admissible paths in time and space, based on information about static obstacles provided by a laser range scanner, as well as motion information, i.e. position and velocity vectors of moving obstacles and the person being followed, provided by the external tracking component [19].

We performed three series of experiments with this system within the robot experimentation hall and the hallways of the FKIE building (cf. Fig. 3). The first series evaluates the robustness of the component-based tracker compared to simpler feature-based techniques. In the second series, the robot autonomously controls the camera to track a target person while other persons are moving around in the field of view of the robot and try to distract it. In the third series, the robot uses the people tracker to autonomously follow a person.

All series were performed during normal working hours with people walking around. The lighting conditions varied strongly during the experiments: some areas show natural daylight, others artificial light. In some parts, the light was switched off resulting in poorly illuminated areas. These conditions resulted in several images with very poor quality. Furthermore, after quick camera movements the camera was out of focus for some frames and capturing images was sometimes delayed resulting in large changes between consecutive frames. To evaluate the tracking, we counted the number of detec-

Fig. 4 Some tracking results. Green points: particles that matched to target; cyan points: particles that didn't match. Rectangles show estimated target state. Yellow rectangle: more than 30% of particles match, otherwise the rectangle is blue.

tions manually. A detection occurs if the center of the target state was on the person[3]. In Fig. 4 we display some of the tracking results.

5.1 Experiment 1: Comparison with Other Feature-based Techniques

Most similar to the here presented approach are color-based trackers. Here, we compare our approach to three other color-based tracking methods. The first is the Camshift tracker [7] based on the MeanShift algorithm [11]. It is a statistical method of finding the peak of a probability distribution, usually obtained with a color histogram. Additionally to the implementation based on the HSV color space that is available from the OpenCV library[4], we used it with two other color spaces: RG chromaticity space and LAB space.

The second and the third method are both based on particle filters. The second approach is a standard method based on color histograms and was implemented according to [27]. The third approach that we call ROI (region of interest) tracking is a simplified version of the here presented method. It uses the same feature maps as in sec. 3.1 but no components. Instead, it considers the whole target region and computes a descriptor based on the ratio of the mean of the target region and the mean of the background as in eq. 2. Thus, it computes a 6-dimensional target descriptor.[5]

To be able to compare the approaches on the same data, several image sequences were acquired by tele-operating the robot and processed offline. We tested 5 different runs, each covering one circle through the hallways (approx. 160 m per run). Each run was performed with a different person as target, with different clothing (cf. Fig. 5). The runs consisted of 1000–1600 frames each. The results are displayed in Tab. 1. In all cases, the component-based tracker performed best, with an average detection rate of 90%. The simpler ROI tracking achieved 77% on average. The approaches based on color histograms (Camshift and histogram with particles) approaches perform considerably worse (33, 45, 40%, and 37%). This is mainly due to problems with illumination changes. For all approaches it turned out that the clothing of the person made a strong

[3] This approximation is actually too optimistic since the region might include a part of the background and still have its center on the target. It is reasonable here anyway since the center is the point the robot uses as target direction.

[4] OpenCV library: http://opencvlibrary.sourceforge.net/ For Camshift, it is usually necessary to adapt the parameters newly for each object. This is difficult for targets like persons which vary strongly in appearance due to different clothing. Since our tracker is applicable to different objects without adapting parameters, we used the Camshift algorithm with the standard parameter set of the OpenCV implementation for all test sequences to make the approaches comparable.

[5] We used almost the same method in [16], but omitted here the orientation features to make the approach comparable to the other methods which are purely color-based.

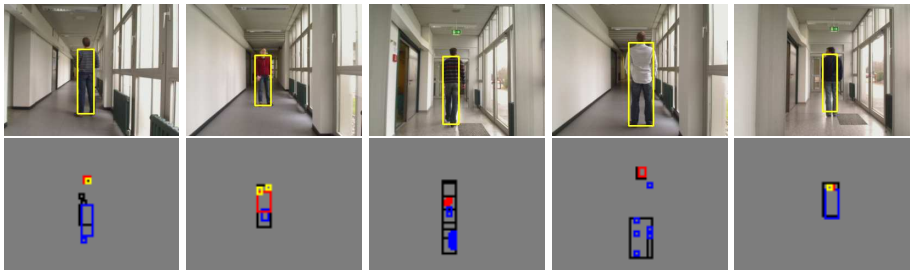

Fig. 5 Experiment 1: Top: initial frames and target regions $\mathbf{R}^*$ (yellow rectangles) used to learn the appearance of the 5 persons. Bottom: the templates $\mathbf{M}_{\mathbf{R}^*}$ that are determined for each of the targets. Each rectangle represents an $m_{i,j}$, its color represents the feature map it was extracted from.

	# Frames	correct detections [%]					
		Cam (HSV)	Cam (RG)	Cam (LAB)	Hist.	ROI	component
1	1477	51	88	39	42	85	95
2	1158	53	62	54	73	98	94
3	1596	5	28	50	17	60	85
4	1392	13	1	10	15	61	90
5	1519	46	47	46	38	80	84
	Average	33	45	40	37	77	90

Table 1 Experiment 1: Comparison of Camshift tracking with three different color spaces (HSV, RG, LAB), color histogram tracking with particles, ROI tracking, and our new component-based tracking. The rows show the results for the 5 persons in Fig. 5.

difference in performance: the larger the contrast and difference to the background, the easier the tracking.

5.2 Experiment 2: Tracking with Autonomous Camera Control

In the 2nd series of experiments, the robot was not moving itself, but autonomously controlled its camera to keep the target person in the center of the frame. We performed 4 runs with 4 different target persons. During all runs other persons were walking in the same area, occasionally occluding the target (cf. Fig. 3, 3rd col., and Fig. 4, a,b).

This experiment demonstrates the robustness of the tracking mechanism and especially the ability to discriminate individual persons. The results are shown in Tab. 2. Images in which the target was not visible were not considered for the detection rate but are mentioned in col. 4. It shows that the tracking works generally very well, the average detection rate is 91%. Most difficulties occurred in example 4, since here two people were sometimes confused.

5.3 Experiment 3: Autonomous Person Tracking

In the 3rd series of experiments, the robot followed a person autonomously. Three runs were performed in the robot experimentation hall and another four in the hallways of FKIE (cf. Fig. 3 and Fig. 4, c-e). The robot estimated the position of the person in each

	# Frames	detections [%]	# frames without target
1	278	91	0
2	509	92	9
3	437	99	0
4	491	82	0
Average	429	91	2.25

Table 2 Experiment 2: Results of component-based tracking on a stationary robot with autonomous camera control and several people walking around.

	# Frames	detections [%]	# frames without target
1	431	93	1
2	472	96	0
3	560	96	75
4	1533	88	13
5	1199	94	0
6	1612	95	8
7	1116	99	0
Average	989	94	14

Table 3 Experiment 3: Component-based tracking in online experiments used to autonomously drive a robot.

frame and drove autonomously into the direction of the estimated target state.[6] The camera was again controlled to center the target in the frame. The results are displayed in Tab. 3. In all of the runs, the detection rate was above 80%. The robot managed to keep the target person in its field of view very well. If the person was lost by the tracker, an audible signal told the person that it should wait for the robot to catch up again. One example in which the person was lost since it was too far away from the robot is displayed in Fig. 4 e. On the four runs through the hallways the sharp corners were the biggest challenge for the system. The 5th run was aborted on such a corner, because the robot lost the person and then was not sure enough if it detected the right person again. The average detection rate was 94%, showing that a robot equipped with the component-based tracker is able to follow a person autonomously.

6 Conclusion

In this paper, we have presented a component-based approach for visual tracking. We have applied the method to person tracking on a mobile platform which is especially challenging due to real-time constraints, a moving camera, and strong illumination and viewpoint changes. The appearance of a person is learned from an initially provided target region and the resulting target descriptor is used to search for the target in subsequent frames. Advantages of the system are that it determines automatically the most descriminative parts of a target, that it considers not only the appearance of the target but also of the background, and that it is quickly adaptable to a new target without a time-consuming learning phase.

[6] Here, control of the distance to the person is left to the laser-based collision avoidance. The robot approaches the person until a certain minimal distance is achieved.

We showed that the system is able to distinguish individuals and can follow a person autonomously through an environment. However, the task of person tracking in natural conditions is very challenging and there are still settings in which the system has difficulties. Persons with clothing similar to the background (especially camouflage), bright sunlight, and crowded environments are settings in which most systems fail. Adding additional features, e.g. motion cues, and asking for feedback from the target person in cases of ambiguity might help to tackle such problems. There are also cases in which the current approach has difficulties if the appearance of target and background change strongly, e.g. due to strong illumination changes. We are currently working on automatically detecting such changes and adapting the target descriptor accordingly.

References

1. M. Andriluka, S. Roth, and Schiele B. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.
2. K. O. Arras, O. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2D range data. In *Proc. Int'l Conf. on Robotics and Automation (ICRA'07)*, Rome, Italy, 2007.
3. K.O. Arras, S. Grzonka, M. Luber, and W. Burgard. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *Proc. of Int'l Conf. on Robotics and Automation*, 2008.
4. N. Bellotto and H. Hu. Multisensor data fusion for joint people tracking and identification with a service robot. In *Proc. of the IEEE Int'l Conf. on Robotics and Biomimetics*, Sanya, China, 2007.
5. M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *Int'l Journal of Robotics Research*, 24, 2005.
6. N. Beuter, O. Lohmann, J. Schmidt, and F. Kummert. Directed attention - a cognitive vision system for a mobile robot. In *18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009.
7. G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, 1998.
8. C. Breglera, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *Int'l Journal of Computer Vision (IJCV)*, 2004.
9. K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part II: Applications to human modeling and markerless motion. *Int'l Journal of Computer Vision (IJCV)*, 2005.
10. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, 2002.
11. D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. Proc. Conf. Computer Vision and Pattern Recognition, 2000.
12. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.
13. S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. PhD thesis, University of Bonn, Germany, July 2005. Published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer Verlag.
14. S. Frintrop and M. Kessel. Most salient region tracking. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA'09)*, Kobe, Japan, 2009.
15. S. Frintrop, M. Klodt, and E. Rome. A real-time visual attention system using integral images. In *Proc. of Int'l Conf. on Computer Vision Systems*, 2007.
16. S. Frintrop, A. Königs, F. Hoeller, and D. Schulz. Visual person tracking using a cognitive observation model. In *ICRA Workshop on People Detection and Tracking*, 2009.
17. D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. *Int. Conference on Computer Vision (ICCV)*, 1999.
18. Helmut Grabner and Horst Bischof. On-line boosting and vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
19. F. Hoeller, D. Schulz, M. Moors, and F. E. Schneider. Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps. In *Proc. of the Int'l Conf. on Robots and Systems (IROS)*, pages 1260–1265. IEEE, 2007.

20. M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int'l J. of Computer Vision (IJCV)*, 29(1):5–28, 1998.
21. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
22. B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int'l J. of Computer Vision, Special Issue on Learning for Recognition and Recognition for Learning*, 77(1-3):259–289, 2008.
23. T. Mathes and J. H. Piater. Robust non-rigid object tracking using point distribution manifolds. In *Proc. of the 28th Annual Symposium of the German Association for Pattern Recognition (DAGM)*, 2006.
24. I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *Int'l Journal of Computer Vision*, 53(3), 2003.
25. M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Int'l Conf. on Robotics and Automation (ICRA)*, 2002.
26. Stephen E. Palmer. *Vision Science, Photons to Phenomenology.* The MIT Press, Cambridge, MA, 1999.
27. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. Proc. of European Conf. on Computer Vision, 2002.
28. P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proc. of the IEEE*, 92(3), 2004.
29. K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP – Image Understanding*, 59(1):94–115, 1994.
30. D. Schulz. A probabilistic exemplar approach to combine laser and vision for person tracking. In Proc. of the Int'l Conf. on Robotics Science and Systems, 2006.
31. D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. Int'l Journal of Robotics Research, 22(2), 2003.
32. Xuan Song, Jinshi Cui, Hongbin Zha, and Huijing Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *European Conference on Computer Vision (ECCV)*, 2008.
33. F. Tang and H. Tao. Object tracking with dynamic feature graph. In *Proc. of the IEEE Workshop on VS-PETS*, 2005.
34. G. Taylor and L. Kleeman. A multiple hypothesis walking person tracker with switched dynamic model. In *Conf. on Robotics and Automation (ACRA)*, 2004.
35. A. Tiderko, T. Bachran, F. Hoeller, D. Schulz, and F. E. Schneider. RoSe – a framework for multicast communication via unreliable networks in multi-robot systems. *Robotics and Autonomous Systems*, 56(12):1017–1026, 2008.
36. K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int'l Journal of Computer Vision (IJCV)*, 48(1):9–19, 2002.
37. R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding (CVIU), special issue Modeling People*, 2006.
38. Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision (IJCV)*, 75(2), 2007.

Simone Frintrop got her diploma in computer science in 2001 from the University of Bonn, Germany, was a Ph.D. student at the Fraunhofer institute AiS in St. Augustin, Germany, and got her Ph.D. in 2005. 2005-2006 she was a postdoctoral researcher at the Royal Institute for Technology (KTH) in Stockholm, Sweden. Currently, she works as Senior Scientist in the Intelligent Vision Systems Group at the University of Bonn, where she investigates cognitive methods for intelligent vision systems.

Achim Königs received his diploma in computer science in 2008 from the University of Bonn, Germany. Currently, he works as Ph.D. student at the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), where he investigates intelligent vision systems in conjunction with unmanned systems.

Frank Hoeller finished his diploma thesis in computer science in 2006 at the University of Bonn, Germany. Currently, he works as Ph.D. student at the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE). His main studies concern autonomous navigation and planning for unmanned systems.

Dirk Schulz received his Doctorate degree in Computer Science from the University of Bonn in 2002. In 2003 he worked as a postdoctoral researcher at the University of Washington in Seattle, USA. From 2004 to 2007 he was a postdoctoral researcher at the Computer Science department of the University of Bonn. Since 2008 he is head of the Unmanned Systems group at the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE) in Wachtberg, Germany. His primary research interests include networked multi-robot systems as well as state estimation and sensor fusion techniques with applications in robotics and intelligent environments.